AMENDMENTS TO THE SPECIFICATION:

Page 1, immediately preceding the paragraph commencing at line 3 ("The present invention relates to a..."), insert the following heading and sub-heading:

BACKGROUND

1. Technical Field

Page 1, immediately preceding the paragraph commencing at line 5 ("A single computer may have many..."), insert the following sub-heading:

2. Related Art

Please amend the paragraph beginning at page 1, line 22, as follows:

A paper entitled "System-Managed Storage", by J. Gelb, in the IBM Systems Journal; 1989; 28, 1; p77, proposes the use of software to overcome problems caused by users having control over where in the network data is placed. Instead of relying on a user to choose a memory device for storing his data, a user indicates a desired characteristic of the storage - e.g. "STORCLAS = CRITICAL" and a computer process determines a suitable memory device for storing the file. The system ~~adminstrator~~ administrator provides mapping data matching storage devices to desired characteristics of storage. That mapping data is used by a storage management

program to select a physical device for storing a file on the basis of the desired characteristic(s) of storage provided by the user.

Page 2, immediately preceding the paragraph commencing at line 29 ("According to a first aspect..."), insert the following heading:

BRIEF SUMMARY

Page 5, immediately preceding the paragraph commencing at line 9 ("In order that the present invention may be better..."), insert the following heading:

BRIEF DESCRIPTION OF THE DRAWINGS

Page 6, immediately preceding the paragraph commencing at line 6 ("Figure 1 illustrates an internetwork..."), insert the following heading:

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

Page 6, paragraph commencing at line 10:

Attached to the fixed local area network 50 are a server computer 12, and five desktop PCs (10,14,16,18,20). The first wireless local area network 60 has a wireless connection to a first laptop computer 26 and second laptop computer 28[[,]]; the second

wireless local area network 14 has wireless connections to a third laptop computer 24 and a personal digital assistant (pda) 22.

Page 6, paragraph commencing at line 16:

Also illustrated is a CD-ROM [[16]] 40 which carries software which can be loaded directly or indirectly onto each of the computing devices of Figure 1 (12 – 28) and which will cause them to operate in accordance with a first embodiment of the present invention when run.

Page 7, paragraph commencing at line 16:

The CD-ROM [[16]] 40 contains client and host application programs and other classes written in the Java programming language. Each of the classes and the client application program are installed and run on one of the computers (for example, on the PC 10). Each of the classes and the host application program is installed and run on the other computers. As will be understood by those skilled in the art, the application programs and other classes provided on the CD-ROM utilise classes provided as part of the DIET agents platform software.

Page 7, paragraph commencing at line 16:

The application program and classes provided on the CD-ROM [[16]] 40 are illustrated in Figure 3. Also shown in that diagram are some of the important interactions between the different classes.

Page 10, 1st paragraph:

HostResourceDiscoveryStrategy: On receiving a DiscoveryMessage from [[an]] a DietHostScoutEcho agent, this adds the computer from which the agent came to the list of suitable hosts for storing a fragment of the file. It keeps count of the number of suitable hosts so far indicated, and when that number is equal to the "token bucket" associated with the distributed file storage, this [[calls]] is called a sampleHosts() method of the SlaveJob class, passing it the "token bucket" value, the address of the client computer, and an array of addresses of computers that have passed the test for storing a fragment of the file.

Page 11, 4th full paragraph:

[[An]] A more advanced process of finding suitable storage locations for the file fragments will now be illustrated with reference to Figures 5 to 8. In this case, the use of agents to transfer messages between the different software components is not described - however, it is to be understood that such a mechanism for communicating

information between classes could be used. Alternatively, the communication between the software components could take place using standard Java techniques for such communication such as Remote Method Invocation. This example supposes that PC 10 seeks to discover six on-line devices capable of storing one fragment each.

Page 12, last paragraph:

The above software forms a mechanism that might be used in a distributed storage system or decentralised file system. An example of such a decentralised file system is described in Robertson, D. et al, "Persistent, Reliable, Decentralised File System - DFS" presented at the London Communications Symposium 2002. As mentioned in [[the]] that paper, the file fragmentation methods could use a number of different erasure codes. Example erasure codes which could be used are described in N. Alon, J. Edmonds, M. Luby, "Linear Time Erasure Codes With Nearly Optimal Recovery", Proc. of the [[36 th]] 36th Annual Symp. on Foundations of Computer Science, 1995, pp. 512-519, and a paper by John Byers, Michael Luby, Michael Mitzenmacher entitled *"Accessing Multiple Mirror Sites in Parallel: Using Tornado Codes to Speed Up Downloads"* which appears as **ICSI Technical Report TR-98-021**, July 1998, and is currently available for viewing at the following URL: http://www.icsi.berkeley.edu/~luby/PAPERS/mirrdown.ps. The tornado code is especially preferred because of the improved coding and decoding speeds.

Pages 15-16, bridging paragraph:

Thus, referring to Figure 5, together with Figure 9 which is a flowchart illustrating the processing performed by each Discovery Strategy, it can be seen that in the present example, after receiving a message at step S5, the device 10 determines that it is the originating device at step S10 and proceeds to step S15 in which it updates its second hash-map table by noting that it has three neighbours stored in its first hash-map table, namely server 12, desktop computer 20 and laptop computer 26, with each neighbour having an (initial) equal probability of 1/3. The Discovery Strategy then determines that all three of its neighbours are on-line and thus calculates normalised probabilities which are again therefore each equal to 1/3. Since the token bucket value is six, it generates one message to each of its neighbours with each message having a token bucket value of 2 (where a perfect distribution is not possible a weighted random selection is made instead, [[eg]] e.g. if the token bucket value were 5, one of the neighbours would have been selected at random whose message would have a token bucket value of 1, with the other two having values of 2 each, etc.). The Discovery ~~strategy~~ Strategy then makes a record of which neighbours it has sent tokens to and returns to step S5 to await receipt of a further message.

Pages 17-18, bridging paragraph:

Referring again now to Figure 6, together with Figure 8, it can be seen that in the present example when device 12 receives the request message indicated by arrow 112, it firstly establishes (at step S10) that it is not receiving an originating request message since the previous address and client address fields are non-null (they have device 10's address); it also checks that it is a new message (at step S20) by checking its records and noting that the file-identifier is new to it, whereupon it decrements (at step S25) the number of hops field (in this example, say from 5 to 4). The Discovery Strategy of device 12 then checks (at step S30) to see if it satisfies the conditions set out in the test field (in this example, whether it has disk space available for storing 100 Kbytes), it determines that it does satisfy this condition and thus sends (at step S35) an acceptance message to the client host device 10 and then decrements (at step S40) the token bucket by 1; it notes (at step S45) that the token bucket still has one remaining token to be disposed of; it checks (at step S50) that the number of hops is still greater than zero (in this example, it's now 4); it then determines (at step S55) that it has 3 eligible neighbours onto which it may forward the message, whereupon it updates (at step S65) its second hash-map table and selects at random one of its eligible neighbours (in this case device 14) to which it sends a message indicated by arrow 114 including one remaining token left to dispose of and keeps a record of this before returning to ~~awaiting~~ <u>await</u> a further request message.

Page 18, 1st full paragraph:

Devices 20 and 26 perform similar tasks and thus send on messages indicated by arrows 122 and 128 with one token each to portable digital assistant (pda) device 22 and laptop computer device 28, respectively.

Page 25, top of page, delete "CLAIMS" and insert the following heading:

WHAT IS CLAIMED IS: